Sample-Efficient Neurosymbolic Deep Reinforcement Learning

Extended Abstract

Celeste Veronese
University of Verona
Verona, Italy
celeste.veronese@univr.it

Alessandro Farinelli
University of Verona
Verona, Italy
alessandro.farinelli@univr.it

Daniele Meli
University of Verona
Verona, Italy
daniele.meli@univr.it

ABSTRACT

Reinforcement Learning (RL) provides a standard framework for sequential decision-making, but state-of-the-art Deep RL (DRL) methods are often sample-inefficient and struggle to generalize beyond small-scale training scenarios. We propose a neuro-symbolic DRL approach that integrates background symbolic knowledge to improve sample efficiency and generalization to more complex, unseen tasks. Partial policies learned in simple domains are transferred as logical rules and used for online reasoning to guide learning by biasing exploration and rescaling Q-values during exploitation. This integration enhances interpretability and accelerates convergence, particularly in sparse-reward and long-horizon settings. Experiments show superior performance over state-of-the-art reward machine methods.

KEYWORDS

Neurosymbolic RL; Knowledge Transfer; Sample Efficiency

ACM Reference Format:
Celeste Veronese, Alessandro Farinelli, and Daniele Meli. 2026. Sample-Efficient Neurosymbolic Deep Reinforcement Learning: Extended Abstract. In *Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), Paphos, Cyprus, May 25 – 29, 2026*, IFAAMAS, 3 pages. https://doi.org/10.65109/RPUM9981

1 INTRODUCTION

Deep Reinforcement Learning (DRL) has shown strong potential for solving sequential decision-making problems in complex real-world domains, characterized by high-dimensional observations, multiple objectives, and uncertain dynamics. However, DRL still suffers from *sample inefficiency*, significantly limiting its scalability and generalization in the presence of long planning horizons, sparse rewards, and numerous sub-goals [4]). Existing solutions either rely on extensive prior data [1], human intervention [13], or restrictive assumptions on policy parametrization and regularization [7, 9]. Heuristic-guided DRL approaches, including reward shaping [3, 15], can partially solve sample inefficiency, but remain highly sensitive to heuristic quality and often require many environment interactions [2]. Neuro-symbolic (NeSy) approaches commonly rely on handcrafted logical specifications to define sub-goals or automata, either within hierarchical planning frameworks [8] or via reward shaping [3, 15]. More recent methods attempt to learn symbolic knowledge from agent traces [5, 12, 17], but they typically focus on limited aspects of the learning process, such as exploration alone or tabular initialization, and do not provide a fully integrated and scalable solution for DRL.

In this extended abstract[1], we propose a novel NeSy methodology to improve the sample efficiency and generalization of DRL. Our approach transfers symbolic knowledge, represented as logical rules approximating policies learned in small, easy-to-solve environments, to guide learning in more complex scenarios. Through symbolic reasoning, the agent identifies promising actions given its observations, and this knowledge is exploited directly at the algorithmic level of DRL. In contrast to methods requiring exact sub-goal definitions or perfect heuristics, our framework operates effectively with imperfect symbolic knowledge and does not require policy re-tuning when scaling to larger domains. Focusing on ϵ-greedy DRL algorithms, we jointly influence exploration and exploitation by biasing action selection toward symbolically entailed actions and adaptively rescaling Q-values according to the exploration parameter. This mechanism, combined with an ϵ-decay strategy, enables more efficient early exploration while progressively reducing the influence of symbolic reasoning. Compared to Statistical Relational Learning approaches [6, 11], which struggle to scale due to rigid policy search spaces, our method achieves robust generalization to domains with longer horizons and more sub-goals, where standard DRL algorithms typically fail.

2 METHODOLOGY

Our NeSy DRL strategy combines ϵ-greedy DRL exploration and exploitation with symbolic reasoning over logical knowledge, approximating a good partial policy acquired in small and simple domain instances. Symbolic reasoning requires mapping the MDP domain to the logical knowledge. To this aim, we leverage the procedure in [12, 16, 17], relying on the Answer Set Programming (ASP) formalism [10]. Given the ASP representations of the MDP state features ($\mathcal{F}$) and actions ($\mathcal{A}$), we define a *feature map* $F_{\mathcal{F}} : S \to \mathcal{H}(\mathcal{F})$ and an *action map* $F_{\mathcal{A}} : A \to \mathcal{H}(\mathcal{A})$, where $\mathcal{H}(\mathcal{F})$ and $\mathcal{H}(\mathcal{A})$ denote the Herbrandt bases (i.e. sets of ground terms) of $\mathcal{F}$ and $\mathcal{A}$, respectively. We can then represent the policy's information using $F_{\mathcal{F},A}$, defining a *partial logical policy* $\pi_{ASP} : \mathcal{F} \to \mathcal{A}$. The logical policy encodes normal rules in the form $\mathtt{a} :\!\!- \mathtt{f}_1, \ldots, \mathtt{f}_n$, with $f_i \in \mathcal{F}$, $a \in \mathcal{A}$, and computing the answer set of the theory given the current MDP state returns a set of suggested actions $\mathcal{A}_{\pi_{ASP}}$.

2.1 Neuro-Symbolic Training

Once π_{ASP} is defined, our goal is to reason over it to increase sample efficiency in ϵ-greedy DRL by biasing the agent towards the most

[1] Full paper available at https://arxiv.org/abs/2601.02850

Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), C. Amato, L. Dennis, V. Mascardi, J. Thangarajah (eds.), May 25 – 29, 2026, Paphos, Cyprus. © 2026 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). https://doi.org/10.65109/RPUM9981

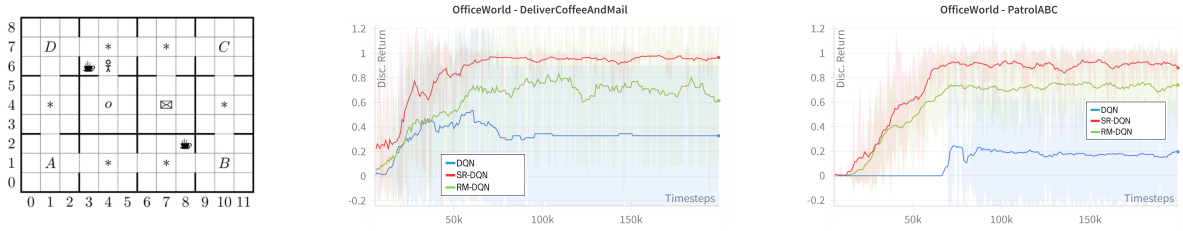

Figure 1: Office domain from [5, 15] (left) and results on the *DeliverCoffeeAndMail* (center) and *PatrolABC* (right) tasks.

promising actions, according to the symbolic policy. For simplicity, we frame our methodology in the context of the DQN algorithm (for which the reader can refer to [14]); however, it can be easily extended to any other ϵ-greedy DRL approach. In both the exploration and exploitation phases, once the current state is observed, automated reasoning is performed over π_{ASP} to identify the set $\mathcal{A}_{\pi_{ASP}}$ of actions (ground terms in ASP formalism) entailed by the background policy knowledge, given the current set of ground environmental features $F_{\mathcal{F}}(s_t)$. Suggested ASP ground actions are then translated to the MDP action space A by considering $F^{-1}_{\mathcal{A}}$, obtaining $A_{\pi_{ASP}}$. Importantly, since the symbolic knowledge could be inaccurate, e.g., learned from previous example executions [5], we also define a confidence level $\rho \in [0, 1)$ about π_{ASP}, that allows us to modulate the impact of the guidance introduced by the additional policy according to how much we trust the acquired knowledge.

Neuro-Symbolic Exploration. In the exploration phase, the agent selects an action from A according to a weighted probability distribution, where the weights are defined for each action as $w_a = \rho$ if $a \in A_{\pi_{ASP}}$, otherwise $w_a = 1 - \rho$, and then normalized, such that $\sum_{a \in A} w_a = 1$. The higher ρ value, the higher the probability that the agent selects an action suggested by background policy knowledge. If $A_{\pi_{ASP}} = \emptyset$ (i.e., π_{ASP} cannot suggest any valuable action at the given state s_t), then a uniformly random action is selected from A as in standard DQN.

Neuro-Symbolic Exploitation. In the exploitation phase, given the current DQN's Q-values, we employ $A_{\pi_{\text{ASP}}}$ to rescale them by a factor $k_a = 1 + (\epsilon * w_a)$ for each action, with w_a determined as in the exploration phase. This adjusts the action values within the context of the most promising action set $\mathcal{A}_{\pi_{\text{ASP}}}$, according to the confidence parameter ρ. Adding ϵ as an additional rescaling parameter allows the agent to increasingly trust the neural network's estimations as the training proceeds (following standard epsilon-decay rules).

3 EMPIRICAL EVALUATION

[2] We evaluate our SR-DQN methodology on the *OfficeWorld* domain, widely used in related literature [5, 15]. The agent has to either visit rooms in a certain order (*VisitAB* and *VisitABC* tasks) or bring objects to the office location (*DeliverCoffee* and *DeliverCoffeeAndMail* tasks), depending on the task. Crucially, to test the scalability performance of SR-DQN, we employ the policy learned by [5] in the *DeliverCoffee* and *PatrolAB* tasks as partial policies in the more complex *DeliverCoffeeAndMail* and *PatrolABC* tasks, respectively. In this way, we assess the sampling efficiency of our methodology when generalizing to longer planning horizons and more sub-goals. more specifically, we employ the following π_{ASP} for the *DeliverCoffeeAndMail* task (Rules (1) and (2)) and *PatrolABC* task (Rule (3) and Rule (4)):

```
goto(X) :- coffee(X), not hasCoffee, not hittingDecor.   (1)
goto(X) :- office(X), hasCoffee, not hittingDecor.       (2)
goto(A) :- not visited(A), not hittingDecor.             (3)
goto(B) :- visited(A), not hittingDecor.                 (4)
```

where $\mathcal{A} = \{\texttt{goto(X)}\}$ denotes the action of moving to item `X`. The above specifications suggest that the agent should first pick up the coffee and then reach the office (Rules 1, 2). Similarly, Rules 3 and 4 state that the agent should first visit room A and then visit room B. In both cases, the agent should not hit any decoration.

3.1 Scalability Study

We compare SR-DQN against the performance of a standard DQN algorithm and DQN with reward machines (RM-DQN in the figures) as designed in [15]. For each method, we evaluate the discounted return. For reward machines, we test different rewards for state transitions and keep the best-performing ones in the tuning scenarios (excluding additional rewards in the plots for fair comparison). Since we do not have information about the confidence level of [5], we empirically set $\rho = 0.8$. Figure 1 shows the performance of SR-DQN and the baselines. For both tasks, we tuned the base DQN algorithm to solve the easier setting (i.e. *DeliverCoffee* and *PatrolAB*) and then used the same set of hyperparameters to train all the agents in the more challenging tasks. On average, SR-DQN achieves the highest return by the end of training, also proving to be more stable with a lower standard deviation with respect to DQN in particular. On the other hand, RM-DQN converges more slowly to a lower average return with larger variance.

4 CONCLUSION AND FUTURE WORK

We presented SR-DQN, a novel neuro-symbolic DRL approach addressing scalability and sample inefficiency in environments with long planning horizons, sparse rewards, and multiple sub-goals. Our method leverages partial logical policy specifications learned in easy-to-solve domains and uses automated reasoning to infer promising actions, biasing both exploration and Q-values in ϵ-greedy DRL. An ϵ-decay schedule balances symbolic reasoning and neural learning over time. Experiments in *OfficeWorld* demonstrate that SR-DQN consistently outperforms selected baselines. Future work will extend the approach beyond ϵ-greedy DRL algorithms and towards more expressive policy representation.

[2] Code available at https://github.com/Isla-lab/sample_efficient_nesy_drl

REFERENCES

[1] Martin Bertran, Natalia Martinez, Mariano Phielipp, and Guillermo Sapiro. 2020. Instance-based generalization in reinforcement learning. *Advances in Neural Information Processing Systems* 33 (2020), 11333–11344.

[2] Ching-An Cheng, Andrey Kolobov, and Adith Swaminathan. 2021. Heuristic-guided reinforcement learning. *Advances in Neural Information Processing Systems* 34 (2021), 13550–13563.

[3] Giuseppe De Giacomo, Luca Iocchi, Marco Favorito, and Fabio Patrizi. 2019. Foundations for restraining bolts: Reinforcement learning with LTLf/LDLf restraining specifications. In *Proceedings of the international conference on automated planning and scheduling*, Vol. 29. 128–136.

[4] Gabriel Dulac-Arnold, Nir Levine, Daniel J Mankowitz, Jerry Li, Cosmin Paduraru, Sven Gowal, and Todd Hester. 2021. Challenges of real-world reinforcement learning: definitions, benchmarks and analysis. *Machine Learning* 110, 9 (2021), 2419–2468.

[5] Daniel Furelos-Blanco, Mark Law, Anders Jonsson, Krysia Broda, and Alessandra Russo. 2021. Induction and exploitation of subgoal automata for reinforcement learning. *Journal of Artificial Intelligence Research* 70 (2021), 1031–1116.

[6] Rishi Hazra and Luc De Raedt. 2023. Deep explainable relational reinforcement learning: a neuro-symbolic approach. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 213–229.

[7] Maximilian Igl, Kamil Ciosek, Yingzhen Li, Sebastian Tschiatschek, Cheng Zhang, Sam Devlin, and Katja Hofmann. 2019. Generalization in reinforcement learning with selective noise injection and information bottleneck. *Advances in neural information processing systems* 32 (2019).

[8] Harsha Kokel, Sriraam Natarajan, Balaraman Ravindran, and Prasad Tadepalli. 2023. RePReL: a unified framework for integrating relational planning and reinforcement learning for effective abstraction in discrete and continuous domains. *Neural Computing and Applications* 35, 23 (2023), 16877–16892.

[9] Kimin Lee, Kibok Lee, Jinwoo Shin, and Honglak Lee. 2020. Network Randomization: A Simple Technique for Generalization in Deep Reinforcement Learning. In *Eighth International Conference on Learning Representations, ICLR 2020*. International Conference on Learning Representations.

[10] Vladimir Lifschitz. 1999. Answer set planning. In *Logic Programming and Nonmonotonic Reasoning: 5th International Conference, LPNMR'99 El Paso, Texas, USA, December 2–4, 1999 Proceedings 5*. Springer, 373–374.

[11] Giuseppe Marra, Sebastijan Dumančić, Robin Manhaeve, and Luc De Raedt. 2024. From statistical relational to neurosymbolic artificial intelligence: A survey. *Artificial Intelligence* (2024), 104062.

[12] Daniele Meli, Alberto Castellini, and Alessandro Farinelli. 2024. Learning logic specifications for policy guidance in pomdps: an inductive logic programming approach. *Journal of Artificial Intelligence Research* 79 (2024), 725–776.

[13] DJ Strouse, Kevin McKee, Matt Botvinick, Edward Hughes, and Richard Everett. 2021. Collaborating with humans without human data. *Advances in Neural Information Processing Systems* 34 (2021), 14502–14515.

[14] Richard S. Sutton and Andrew G. Barto. 2018. *Reinforcement Learning: An Introduction* (2nd ed.). MIT Press.

[15] R. Toro Icarte, T. Q. Klassen, R. A. Valenzano, and S. A. McIlraith. 2018. Using Reward Machines for High-Level Task Specification and Decomposition in Reinforcement Learning. In *Proceedings of the International Conference on Machine Learning (ICML)*. 2112–2121.

[16] Celeste Veronese, Daniele Meli, and Alessandro Farinelli. 2025. Learning Symbolic Persistent Macro-Actions for POMDP Solving Over Time. In *Proceedings of The 19th International Conference on Neurosymbolic Learning and Reasoning (Proceedings of Machine Learning Research, Vol. 284)*, Leilani H. Gilpin, Eleonora Giunchiglia, Pascal Hitzler, and Emile van Krieken (Eds.). PMLR, 1026–1040. https://proceedings.mlr.press/v284/veronese25a.html

[17] Celeste Veronese, Daniele Meli, and Alessandro Farinelli. 2025. Online Inductive Learning from Answer Sets for Efficient Reinforcement Learning Exploration. In *Hybrid Models for Coupling Deductive and Inductive Reasoning*, Pierangela Bruno, Francesco Calimeri, Francesco Cauteruccio, and Giorgio Terracina (Eds.). Springer Nature Switzerland, Cham, 93–106.